A Monitoring Approach for Safe IPv6 Renumbering

Frédéric Beck, Isabelle Chrisment, Olivier Festor

A Monitoring Approach for Safe IPv6 Renumbering

Frederic Beck Isabelle Chrisment
Olivier Festor
LORIA - INRIA Lorraine
Campus Scientifique - BP 239
54506 Vandœuvre-les-Nancy Cedex, France
Email: {frederic.beck,isabelle.chrisment,olivier.festor}@loria.fr

Abstract

Network renumbering is a very interesting feature of IPv6. It is also one of the most risky procedure which needs special attention in the management plane. In this paper we identify the challenges of renumbering and demonstrate how monitoring can improve this process. We also present an implementation of a monitoring framework and share the experience gained in its deployment.

1. Introduction

To overcome some of the limits of IPv4, especially the exhaustion of the address space due to the exponential growth of the internet of the last decade, a new version of the IP protocol, IPv6 [4] has been standardized within by the IETF, bringing with it advanced built-in services. Over the last ten years, it has been the focus of many studies, including to extend the management plane.

In the scope of the 6Net IST project[1], we studied the impact of one of the new built-in services brought with IPv6, namely IPv6 network renumbering, on the management plane. This study highlighted a lack in management tools for monitoring such a procedure.

One of the very promising service of IPv6 is the renumbering facility which theoretically enables seamless network renumbering. This service is useful in many situations but the increasing experiences and studies performed on large Ipv6 networks also show that this feature needs to be carefully managed to avoid potential network outage.

In this paper, we present the design of a distributed monitoring service that drastically eases the administrators work in following a renumbering procedure. This architecture has been implemented into the NetSV management framework we developped for specific IPv6 services management. Details of the implementation are also presented in the paper.

The paper is organized as follows. We first give a brief presentation of the renumbering process in IPv6 (section 2). We then identify the problems raised during such a process. In section 4 we motivate the need to monitor the renumbering procedure. In section 5, we present the main components of the architecture. Section 6 presents the NetSV implementation architecture and how it is used for renumbering monitoring. The presented work is summarized in section 7 together with an outlook for future work.

2. IPv6 network renumbering

IPv6 network renumbering is a procedure in which all devices located on a subnet change their IPV6 prefix address. Several investigations have already been successfully made on this topic, leading to one Internet draft [6] and one Request For Comments [3] recently published. The first one acts as a reminder for all the important things when a network is renumbered, the second one describes the different steps of a renumbering procedure. These are our main references, jointly with 6net's related deliverables, D3.6.1 [1] and D3.6.2 [2].

A renumbering can be triggered by many different events, like the uplink prefix changes, mostly because of the migration to a new provider or a modification in its topology, a change in the internal topology, a network merge, dial up connections... The frequency of these events is very variable, and can complicate the task of the administrator. But renumbering a network must avoid long disruptions for the users. The best case would be that they do not notice anything, i.e. that renumbering is completely transparent for them.

Thus, it is very important, before doing a renumbering, to identify the cause and if possible the moment when it will occur in order to anticipate the needed preparations in order

[1] http://www.6net.org

to prevent as many problems as possible. Before performing a network renumbering, the network administrator has to prepare and check the availability of some functionalities and mechanisms :

- Multihoming on hosts' interfaces.
- Renumbering of all the routers and switches.
- Adjusting the prefixes' lifetimes in RAs, DHCP leases, DNS entries validity in caches...

Renumbering an IPv6 network can be done with a planned service outage, but the main idea is to avoid such a thing, and make the transition transparent for the users. To do so, a procedure in 8 steps has been defined :

1. Stable and working situation with an existing prefix which we will call the *old prefix*. IPv6 hosts must be configured to perform Address Auto-Configuration and be able to use multiple addresses on the same interface (multihoming).
2. Obtain the new prefix and new reverse zone from the delegating authority. Assign a sub-prefix from the new prefix to each link. Add DNS entries for the new prefix, adjust lifetimes, and reconfigure DHCP if stateful addressing is used.
3. Set up a parallel routing architecture for the new prefix. Reconfigure switches and routers with the new prefix without advertising it in Router Advertisements (RAs). Update ingress and egress filtering for the new prefix. At the end of this step, the prefix is announced outside the network.
4. Hosts' addressing : the new prefix is announced in RAs. Update of DNS and reverse-DNS entries.
5. Stable configuration where the network is multihomed.
6. Old prefix is obsolete (lifetimes set to zero). Transition from the old to the new prefix for services.
7. When all dependencies to the old prefix have been cleared, remove it from the RAs, the routing architecture, filtering... Addresses for the old prefix are deleted from the hosts' interfaces.
8. Equivalent to the first state, but using the new prefix.

3. Renumbering Problems

Several known elements/practices can lead to unsuccessfull renumbering and both network and service outage. In this section we list them.

First of all, addresses usually appear as "hard coded" parameters in configuration files, and unless these are updated according to the changes, the feature or the tool itself will not be working. After such a modification in the service/tool's configuration, it must usually be restarted in order to reread the configuration file. "Hard coded" addresses do not only appear in tools configuration, they also are present in firewall, routing tables and even sometimes in the applications' source code. Some hosts or routers/switches may be manually configured, and thus, address autoconfiguration will not be performed when the new prefix changes, and the addressing of these hosts will not be updated. This can lead to a situation where the host is not reachable anymore, and then all the associated services will be down. For firewall rules or routing tables, the updates can be done dynamically with shell commands, but in case of source code modification, the application must be rebuilt and the executable replaced.

In the same way, some applications or services use DNS resolutions. To avoid problems, these resolutions must be done as often as possible, which means that the perfect scenario would be the use of systematic resolutions (when taking the assumption that the DNS entries are maintained up-to-date). But in order to save resources and increase the program's rapidity, cache systems are used. If these cache's lifetimes follow the lifetimes of the DNS entries, as the announced lifetimes are shorter during a renumbering, linked problems will not be too much marked. However, problems can appear for applications/services that make a single resolution (at the start up usually), or these which bind to an address modified during the renumbering. In these cases, the service must be restarted.

Some problems are linked to the nature of the applications themselves. Actually, the problem appears for applications which keep stored information concerning the evolution of an host in order to update statistics for example. After a renumbering, depending on their configuration or the way they identify the monitored hosts (problems are raised when the address is used and can be avoided by using a defined hostname), some data continuity problems can appear.

On top of that, when the routing architecture for the new prefix is being set up, the network parts where the routing is working is vulnerable to attacks. To protect them, the first thing to do before setting up this architecture is to update the ingress and egress access lists by including rules for the new prefix, and, of course, not advertise this prefix outside the network before that ! This involves another problem : it is necessary, so that the renumbering proceeds correctly, that the network administrator is well informed before the procedure takes place, in order to have enough time to make the needed preparations, and especially for security matters.

4. The Need for Monitoring

As shown in section 3, renumbering an IPv6 network can be a painful and risky adventure, and triggers several problems. These are linked to the procedure itself, but are also dependent from the frequency and the cause of it. Depending on the size of the renumbered network, these problems can be hard to identify or predict. If a renumbering occurs whereas the network administrator isn't prepared, it can lead to the death of the network, as no preparations can be done (firewalling, DNS, routing). Thus, there is an important need to monitor this procedure. In this section, we will take a closer look at the problems highlighted in section 3, and motivate this affirmation.

As it has been already said, manually configured hosts and equipments do not perform address autoconfiguration. It would be helpful for the administrator to have an overview of its network, to see which hosts have updated their addressing, which have not and which updated it wrongly. This kind of monitoring feature makes also possible to detect the malicious or unawaited emission of RAs, if there is a difference between the addressing of some hosts on the network and their supposed behavior.

On step 6 of the IPv6 renumbering procedure, hosts and applications should choose the right time for them to make the transition. But some services could forbid hosts to perform the transition. If a host is using a Virtual Private Network (VPN) or Network File Systems (NFS), it is preferable to update and restart this service before suppressing the old prefix, to avoid services outage. In the same way, hard coded addresses may appear in other kind of applications, and may need an update and a restart with the new prefix, but do not forbid the host to renumber. These updates are repetitive and very similar ; they can be achieved by scripts to make the renumbering procedure easier and to gain a lot of time. Monitoring a renumbering and triggering the execution of these update scripts accordingly would simplify the administrator's task and reduce the transition's duration.

Finally, the main issue of the transition is this period's duration. It is impossible to predict, as many different factors come into play and are variable from one network to another. This duration must be defined before the renumbering occurs, but it implies to keep the two prefix available for a while, and can lead to a significant overcost.

In the face of a dynamically evolutive situation, with unconstrained state transitions over time, only precise and intelligent monitoring can help the network administrator to adapt this period to what happens in the network, to avoid paying for a longer time than the one needed, or finish the transition too early and lead to services' outage. This monitoring is offered by NetSV whose approach is based on a distributed monitoring framework.

5. A distributed monitoring architecture

Our architecture is composed of three entities : management agents, a renumbering manager and service renumbering probes.

Every single IP device in the network (router, host) hosts a monitoring agent whose goals are :

- to detect the starting of a renumbering phase ;
- to monitor the devices behavior evolution during renumbering ;
- to report this evolution to a management application.

The monitoring agent function hosts an intelligent service which is enable to diagnose whether the device did renumber or why the renumbering of the device was not performed. While renumbering is initially a fully decentralized process, our management architecture can be used to centralize this function. Basically the devices can be set in a mode through the management agents, through which their renumbering is conditioned by the approval of a central management station.

This central management station is the second element of our renumbering management infrastructure. It has in charge of validating the renumbering triggers (router advertisement, administrators orders) and to orchestrate the renumbering process.

To complement the agents and the central manager, we did implement a set of probes whose goal is to permanently monitoring the health of the services while the renumbering process is going on. They are used to detect any failure in services that occur due to prefix change in one of the concerned devices. They are in constant interaction with the manager who uses their reports to enable progress of the renumbering procedure to be made.

6. Implementation Architecture

As shown in section 4, it is important to know when a renumbering occurs, validate the hosts' addressing, the transition and the hosts/services configuration, in order to ensure that the network successfully renumbers. But presently, as no application in the management plane is suitable for performing these operations, we proposed to develop such a tool, so-called NetSV, the IPv6 Network renumbering SuperVision tool.

This tool is intended to fill 2 roles :

- monitor the renumbering and validate the addressing of the monitored hosts,
- make diagnostics on the monitored hosts to prevent and/or detect the problems that could be raised during such an operation.

6.1. General architecture

In this section, we describe precisely how we developed and implemented this tool. It is divided in four building blocks :

- a daemon running on the monitoring host,
- a daemon running on the monitored hosts,
- a WEB interface,
- a diagnostic tool.

The monitored daemon sends information on the local addressing to the monitoring host via UDP messages, and the monitoring daemon can use the same kind of messages to trigger special event on the monitored hosts, such as diagnostics. Using the information sent, the monitoring daemon keeps up-to-date an XML file which is used by the web interface to display information about the monitored hosts. A CGI makes possible to pilot the daemon, for example to trigger a diagnostic on an host or program the monitoring of a renumbering. Finally, independent from all these daemons, a tool taking care of checks and diagnostics on the monitored hosts is available. The interaction between these blocks is shown in figure 1.

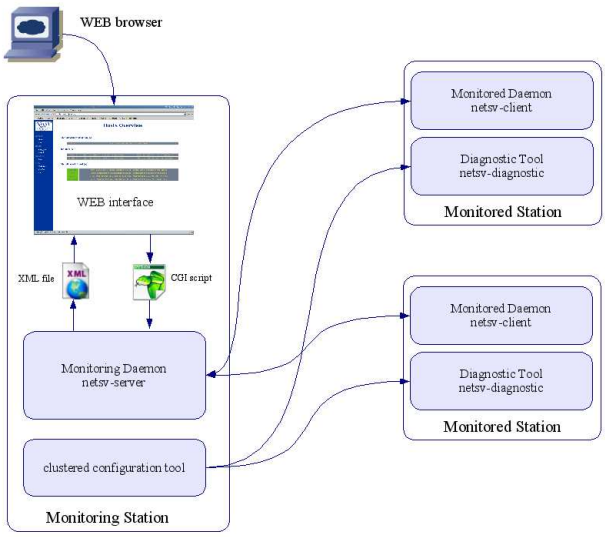

Figure 1. NetSV architecture

6.2. Features description

In this section, we will present the different features and possibilities offered by NetSV, and how they are deployed.

As it has already been said, the agent observes information about the local addressing and the RAs received, and sends them to the manager. Thanks to the data collected, the manager can deduct the network's state, and generates an XML file. The WEB interface uses the XML file generated by the manager and displays the information so that the network administrator can have an overview of the network's state. Via the WEB interface and thanks to a CGI script, it is possible to program the monitoring of a renumbering in the manager. Depending on the data sent by the agents, the manager is able to follow step by step the renumbering procedure, and validate each step before starting the next one.

As the addressing can be followed, it is possible to detect problems in it. The problems identified are what we call *GHOST* and *MISSING* addresses. A *GHOST* address is an address set on an interface whereas no RA is received to announce the corresponding prefix. A *MISSING* address is detected when a prefix is received on an interface but no corresponding address is present on this interface. If one of these cases is met, we can deduce that the host is not performing address autoconfiguration or that a service is running and freezes the addressing. The intelligence in this case is not centralized in the manager, but is distributed in the agents. They take care themselves of comparing their local addressing with the RA received and identify the problems described.

If such a problem (or its resolution) is detected, the agent sends an alert to the manager. Different built-in alter mechanisms are integrated in NetSV. The agent can send an email to a defined Email address, using the Mail Transfer Agent (MTA) running on the monitored host. This implies that this MTA is configured to permit to users or processes on the hosts to send mail to external servers (not only local delivery). Syslog [5] support is also available. All event which trigger a logging can be treated with the syslog daemon from the monitored host. If correctly configured, this daemon can send the alerts to the syslog daemon of the monitoring host. It makes possible to adapt the importance of the alert according to the event itself by tuning the priorities of syslog messages. Finally, NetSV UDP messages are also sent to the manager daemon, so that the problems can be shown on the WEB interface. When a host reported a problem, the indicators concerning this host are displayed in a red colour to inform the administrator in a viusal way, as shown in figure 2.

As described in section 4, the transition's duration is difficult to predict but is a main issue for renumbering. NetSV integrates a monitoring of this transition. When receiving a RA with the lifetimes set to zero for a prefix, it considers that a transition for this prefix begins. Using libpcap [2], it captures the traffic on the interface on which the address

[2] http://www.tcpdump.org

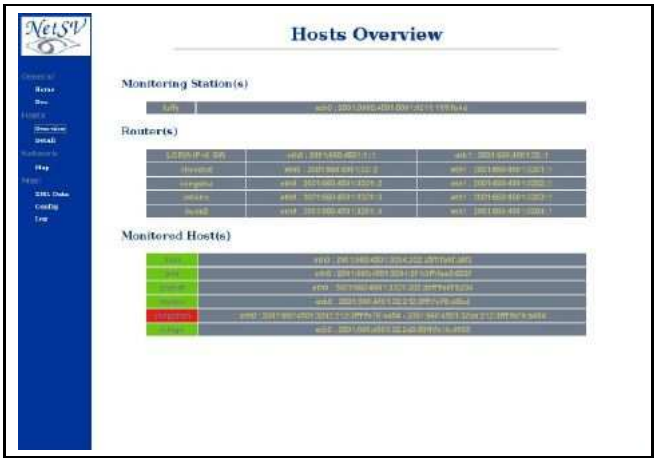

Figure 2. NetSV - Problem in addressing

is set for this prefix. If during a defined period it did not receive any packet destined to the address corresponding to the obsolete prefix, the transition is supposed done for this host. By default, no filter is set, all the packets are analyzed. To monitor specifically some services (HTTP server...), it is possible to define filters by using the same syntax than for tcpdump.

As it enters in the transition period, the agent performs diagnostics for applications and their configuration. These diagnostics can be shell commands or any other program or script. To execute these scripts, the agent uses an XML file, *check.xml*, listing the checks to perform. Each entry follows the given format :

```
<check>
 <type_of_application_tested>
  <instance>
   <category>sub category</category>
   <name>name of application/service tested</name>
   <cmd>command or script to be executed</cmd>
   <output_type>
     int, string or plain_text
   </output_type>
   <condition>
     test to make with the result
   </condition>
   <description>informative text</description>
  </instance>
 </type_of_application_tested>
</check>
```

Several built-in scripts are available, and offer a set of references, which can be adapted by an administrator according to the requirements in network. In the file *check.xml* installed with the tool, all the known tests are included. Here follows an example :

```
<check>
 <running>
  <instance>
   <category>daemon</category>
   <name>sshd</name>
   <cmd>ps axf | grep -v grep | grep sshd: | wc -l</cmd>
   <output_type>int</output_type>
   <condition>more 0</condition>
   <description>session(s) opened on this host</description>
  </instance>
 </running>
</check>
```

This test counts the number of SSH sessions opened on the hosts running the agent. The output of the command is an integer, and it will be compared to the value defined with the tag $< condition >$. If one or more SSH sessions are opened on this host, renumbering the network will be considered *risky* from the point of view of this agent, which can send alerts to the manager, either by Email, or make a window pop up on the host itself.

As these tests are defined in an XML file which is read each time a diagnostic is run, it is possible to dynamically modify it, and the changes will be taken into account by the daemon for the next diagnostic. This can be also triggered by the management plane. By using the WEB interface and the CGI, it is possible to make the manager send a message *DIAGNOSTIC* to an agent. When receiving such a message, the agent launches the diagnostics and sends the result to the manager, and the administrator can interpret it thanks to the WEB interface.

In a same way, a standalone tool is available, which can be combined to a clustered management tool, like cfengine [3], to trigger this diagnostics on several hosts at a same time. If the diagnostics are triggered before the old prefix is announced as obsolete, the required actions can be performed on hosts to avoid problems before they appear.

Finally, DNS is another major issue for renumbering. when a renumbering is performed, the DNS and reverse-DNS entries should be updated according to the new addressing. This update can be simplified by using Dynamic DNS [7], but this does not automate the update. It is mandatory to run the tool *nsupdate* manually on each host when an address appears or is removed. To automate the update, a plugin has been added to NetSV. This plugin consists in a script which is run when a modification in the addressing is detected. This script must be updated by the administrator before deployment to adapt it to the network's configuration :

```
#!/bin/sh

# Variables to update
HOSTNAME=‘hostname‘
SERVER=luffy.loria.fr
DOMAIN=ipv6.renumbering.loria.fr
KEY=/usr/local/netsv/ddns/Kluffy.+157+49723.private

# Delete old entry
echo "server $SERVER" > /etc/nsupdate
echo "prereq yxrrset $HOSTNAME.$DOMAIN IN AAAA"\
  >> /etc/nsupdate
echo "update delete $HOSTNAME.$DOMAIN AAAA"\
  >> /etc/nsupdate
```

[3] http://www.cfengine.org

```
echo "" >> /etc/nsupdate
/usr/bin/nsupdate -k $KEY /etc/nsupdate

# update with addresses
IPADDR=‘/sbin/ifconfig eth0|grep ’inet6’|grep\
  ’Global’|tr -s " "|cut -d/ -f1|cut -d" " -f4‘
echo "server $SERVER" > /etc/nsupdate
echo "prereq nxrrset $HOSTNAME.$DOMAIN IN AAAA"\
 >> /etc/nsupdate
for i in $IPADDR
do
    echo "update add $HOSTNAME.$DOMAIN 60  IN\
      AAAA $i" >> /etc/nsupdate
done
echo "" >> /etc/nsupdate
/usr/bin/nsupdate -k $KEY /etc/nsupdate
```

This script simply removes the old entry and recreates it with the new parameters. However, when performing these operations, the synchronization of the clocks between the DNS server and the monitored hosts has to be maintained, unless the script fails and raises an error.

7. Conclusion and Future Work

Renumbering an IPv6 procedure is a very useful built-in service brought with IPv6, but this procedure can trigger several problems and isn't spontaneous. Thus, monitoring such an operation is important and makes possible to avoid many problems and disturbances.

After highlighting the difficulties related to network renumbering, we presented a tool developed to monitor this procedure. This tool called NetSV is a very good first step. It shows that monitoring an IPv6 network renumbering is possible, and can be a good starting point for further developments.

We are currently working on several extensions to the tool. One of them is its porting to operating systems beyond linux and to provide interoperability with CLI-accessible routers (Quagga, CISCO devices, ...). We are also investigating the use of the tool to support additional service level transition monitoring. During the procedure renumbering, security rules must be updated to run with the new prefix (firewall rules, VPN services, DMZ or extranet renumbering,...). The challenge is to automatically manage this transition phase from security point of view, assuring smooth and correct renumbering

References

[1] 6Net D3.6.1 : Cookbook for IPv6 Renumbering in SOHO and Backbone Networks. http://www.6net.org/publications/deliverables/D3.6.1.pdf.

[2] 6Net D3.6.2 : Cookbook for IPv6 Renumbering in ISP and Enterprise Networks. http://www.6net.org/publications/deliverables/D3.6.2.pdf.

[3] F. Bak, E. Lear, and R. Droms. Procedures for Renumbering an IPv6 Network without a Flag Day. RFC 4192 (Informational), Sept. 2005. Available: http://www.ietf.org/rfc/rfc4192.txt.

[4] S. Deering and R. Hinden. Internet Protocol, Version 6 (IPv6) Specifi cation. RFC 2460 (Draft Standard), Dec. 1998. Available: http://www.ietf.org/rfc/rfc2460.txt.

[5] C. Lonvick. The BSD Syslog Protocol. RFC 3164 (Informational), Aug. 2001. Available: http://www.ietf.org/rfc/rfc3164.txt.

[6] A. F. T. Chown, M. Thompson and S. Venaas. Things to think about when renumbering an IPv6 network - draft-chown-v6ops-renumbering-thinkabout-03, July 2005.

[7] B. Wellington. Secure Domain Name System (DNS) Dynamic Update. RFC 3007 (Proposed Standard), Nov. 2000. Updated by RFCs 4033, 4034, 4035, Available: http://www.ietf.org/rfc/rfc3007.txt.